# Architecture for theatre robotics

Mikko Laamanen

Master's Thesis

ITÄ-SUOMEN YLIOPISTO

School of Computing
Computer Science

February 2015

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu
School of Computing
Computer Science

Laamanen, Mikko Pekka Ilari: Architecture for theatre robotics
Master's Thesis, 51 p.
Supervisor of the Master's Thesis: Ilkka Jormanainen, PhD
February 2015

Educational robotics is widely used in teaching technical subjects, such as computer science and engineering. Our experiences and literature indicate that there would be a need for a truly multidisciplinary educational robotics platform. Theatre robotics aims to fulfill this need by establishing a platform which combines various school subjects.

In this thesis, I will present an architecture for a theatre robotics platform. The research process followed the design research framework, which provides assets for producing practical applications from knowledge. The design of the architecture of the theatre robotics platform is based on experiences from SciKids' technology clubs, which are the main target environments for the platform. The architecture of the theatre robotics platform is designed and evaluated by observing the attributes of chosen approaches.

As a result, this thesis can be used as a guide to implementing the theatre robotics platform in practice. In future, the design will be implemented and improved, and research will continue in developing the theatre robotics platform further.

Keywords: robotics, theatre, educational technology, storytelling, design research

CR Categories (ACM Computing Classification System, 1998 version): A.m, I.2.9, K.3.2

# Preface

The topic of this thesis was born from a need for the designed platform and my desire to write about something practical. I want to thank Professor Erkki Sutinen and Dr. Ilkka Jormanainen for understanding me and providing excellent guidance. I am sorry that this took so long.

I was able to utilize my experiences from working as an instructor in SciKids' technology clubs for several years. I want to thank Joensuu Science Society for orchestrating SciKids' clubs and giving us free hands organizing the club activities. SciKids' has provided challenges, but offered back much more. Thank you, my fellow club instructors for all your efforts. Especially I want to thank Lasse for working together with me to manage the clubs.

Finally, I want to thank my fiancee Hanna. Without you, this thesis would never have been finished. Thank you for pushing me forward.

# List of terms and abbreviations

ACM; Association for Computing Machinery
Android; An operating system for mobile devices
Arduino; An open source prototyping platform
DC-motor; A simple electric motor
Digital storytelling; Storytelling enhanced with digital media
Educational robotics; Educational tools for teaching subject with robots
G-code; Language for controlling automated machinery
IDE; Integrated Development Environment, a software development tool
IEEE; The Institute of Electrical and Electronics Engineers
JSON; JavaScript Object Notation, a structured language for data delivery
LED; Light-Emitting Diode
Lego Mindstorms; An educational robotics set by Lego
Microcontroller; A small processing unit with inputs and outputs
OS; Operating System
PCB; Printed Circuit Board
Robotic storytelling; Digital storytelling enhanced with robotics systems
SciKids'; An after school technology club for children
SDK; Software Development Kit
Servo; A DC-motor with integrated controlling mechanisms
Theatre robotics; Multidisciplinary educational platform

# Contents

# 1 Introduction

Embedded systems and robotics are everywhere around us today [1]. Automation is widely used in industry and embedded systems can be found from modern cars to smart homes, almost from anywhere [18]. The fast development of digital technology leads also into rising expectations on the educational field. Computing education (CE) is in change and requires more competence from teachers and learning environments [28].

For example, England is adapting new curricula in which students will start studying programming from the first grades. Computing will be present thorough all the grades, moving from the basics of programming into more complicated topics, such as sorting algorithms. [3]

Educational robotics is nowadays a trending tool in teaching. Various robotics sets can be found with relatively affordable prices even from local supermarkets. For example, Lego Mindstorms EV3 robotics set can be purchased with approximately 400 €. Educational robotics is usually used to teach especially natural sciences such as physics, mathematics, computer science, and engineering. Williams [31] claims that using educational robotics for teaching computer engineering is qualitatively effective. Then again, educational robotics is rarely applied with humanistic subjects, but experiences from technical classes implicate that there might be a real benefit in doing so. [23, 31, 32]

In this thesis, I approach educational robotics with a multidisciplinary agenda and process the concept of *theatre robotics* introduced by Jormanainen et al. [28]. Briefly described, theatre robotics is a framework with which students can build robots capable of acting out theatre performances. The framework will provide an option for true collaboration between technical and humanistic subjects, like engineering and history. [28]

The main purpose of this thesis is to present an architecture for the theatre robotics platform. The aim is to develop a new, interesting, and efficient way of using robotics in education. As a result, this thesis should be usable as a guide for implementing the theatre robotics platform. I have chosen this subject because, at the moment of writing this thesis, platforms of this kind do not exist, but there seem to be a need for one.

This thesis is organized as follows. In the next sections, the problem definition is stated and the methodology of this study is described. In Chapter 2, the most essential terminology is opened with the literature review. Experiences from children's technology club activities are described in Chapter 3 to support the design process. Section 4 is more technical and forms the actual body of discussion. In the last Chapters, 5 and 6 the design's accountability is discussed and the final conclusion is drawn.

## 1.1 Problem definition

The main goal of this thesis is to *develop an architecture for implementing a theatre robotics platform to be used in educational situations*. This is done by observing different possibilities and aspects for building such platform. The main problem can be divided into four questions:

- **Q1:** Terminology: What does the term *theatre robotics* mean?

  To build a platform for creating theatre robotics applications, the related subjects and terminology need to be first researched and defined.

- **Q2:** Target environment: Which kind of environment the platform should be designed for?

  There needs to be boundaries of some kind for the platform. In which kind of environments should the platform work?

- **Q3:** Design: What are the requirements for the platform in a sense of hardware and software?

  This is the main focus of this thesis. The aim is to combine all required parts to form a robust platform for the needs of theatre robotics.

- **Q4:** Evaluation: Does the complete design meet the requirements?

  The means to define if the platform is suitable for its purpose.

## 1.2 Methodology

My research is built on the design research framework, which is used especially on the educational research field. In design research, products, processes, programs, or policies are developed iteratively to provide a setting for scientific intervention and inquiry.

Design research aims to solve problems by putting knowledge to use and generate new knowledge in the process. The general design research framework constitutes of cycles of analysis and exploration, design and construction, and evaluation and reflection. Outputs of the cycles provide ground for the next phases. [16]

The aim of this thesis is to produce an architecture design for the theatre robotics platform using experiences and observations from target environment. The design science framework is well suitable for the research process for this goal. The research process of the theatre robotics platform followed the design research model presented in Figure 1. The architecture design was produced with one cycle of the design research framework. Following implementation of the design will append more cycles to the whole theatre robotics development process, but that is not in the scope of this thesis.

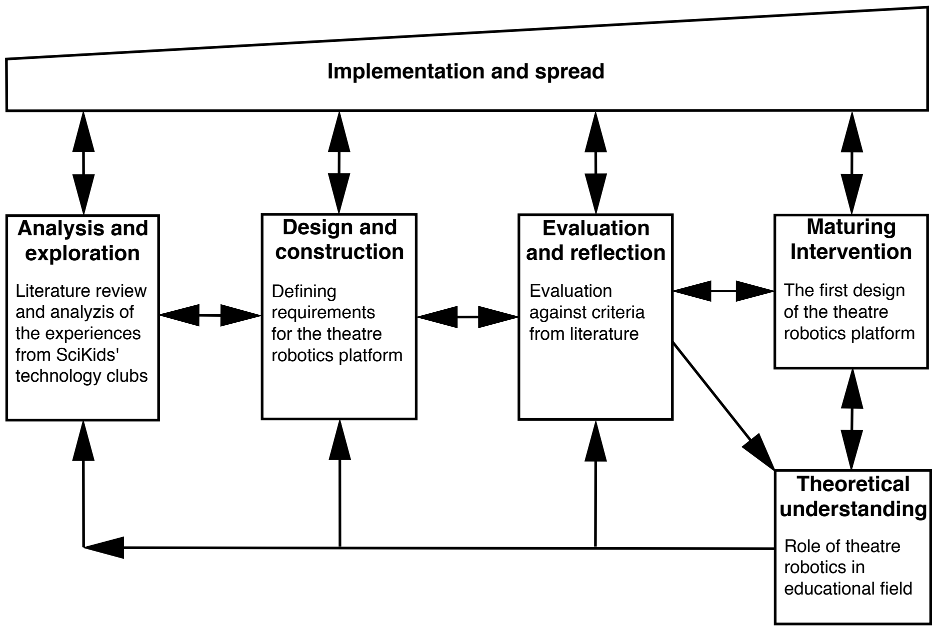


Figure 1: Applied design research process (modified from [16])

The used research process can be divided into the following steps:

- 1. Defining the research problem
- 2. Literature review (**Q1**,**Q2**)

- 3. Description of background studies (**Q2**)
- 4. Listing requirements and creating a design for the platform (**Q3**)
- 5. Evaluating created design (**Q4**)
- 6. Conclusion

Steps 1-3 fall into the analysis and exploration category. The purpose of that phase was to establish ground for the next step (4), developing the requirements for the design and the design itself. Step 5 and 6 implement the evaluation and reflection part of the framework.

Analysis and exploration started with a discussion about the topic with my supervisor. The need for a new platform was supported by my own experiences from working as an instructor in children's technology clubs. The topic of my thesis formed from the practical needs and the results are meant to be used in future as a guide for the implementation of the designed platform. The next step was doing essential background research in form of a literature review. Literature review was done to study existing solutions, the field in general, and to open up the related terminology. I wrote down and analysed my own experiences from working with children in different SciKids' technology clubs for several years. SciKids' provided insight into the use of educational technologies in children's after school technology clubs. Experiences from the clubs were documented by taking pictures and notes of the processes and the results of the clubs. Studied SciKids' clubs were the following. These cases will be described deeply in Chapter 3.

- SciKids' Joensuu, Finland, 2011-2014
- SciKids' Kampala, Uganda, 2013
- SciKids' Experts, Finland, 2013-2014
- SciKids' Theatre robotics, Finland, 2014

Using experiences from the SciKids' as a backbone of this thesis was a natural approach, because my aim was to design the platform for similar environments of this kind, and provided a good ground for design and construction. The experiences from these clubs were discussed and reflected with the literature and as a result, I managed to enlist the essential implications for the theatre robotics. The actual platform design was built with the help of these implications and experiences. The experiences also supported creating a new platform, because previously used technologies did not quite fit in with the formed idea of the theatre robotics platform.

Finally, evaluation and reflection were done by evaluating the created design against criteria from literature and by concluding the gained results. The utilized evaluation criteria were used in similar projects and was therefore applicable to the theatre robotics platform. The conclusion summarizes the process and reflects the gained results against the set goals.

# 2 Background

Answering **Q1** and **Q2** required studying the central topics by reviewing literature from the associated fields. A literature review was done by going through related articles from Association for Computing Machinery's (ACM) and The Institute of Electrical and Electronics Engineers's (IEEE) digital libraries. In search, following key words were used:

- creative robotics
- educational robotics
- creative technology
- theatre robot

Search was focused on articles published between 2004 and 2014. Related articles were selected and will be referred to later in this thesis.

In the next sections, I will open up the most central terms for this work. First, the concept of educational robotics is reviewed. The second section discussed the main concepts of theatre; what kind of aspects affect designing the theatre robotics platform. Finally, I define what is meant by theatre robotics in this work.

## 2.1 Educational robotics

A robot is a device capable of doing autonomous tasks for which it is built and programmed for. Robots can be encountered in various environments and performing a wide range of tasks. Robots are commonly used in factoring products, as tools for accessing difficult places remotely, and also in teaching. [18]

Robotics is adapted to the field of education in the form of educational frameworks and toolkits called educational robotics. Educational robots are tools for teaching different subjects and topics in a contextualized way. Using educational robotics has gained popularity in technical subjects like computer science, engineering, and physics. Educational robotics has been used in all school levels from elementary schools to universities. [32]

A study by Tsang et al. [29] supports benefits of using educational robotics with computer science students. They studied how using Lego Mindstorms robotics kit affects undergraduate students' engagement in learning Java programming language. According to the study, students found learning Java with robotics as a dynamic, interesting, and creative tool. Especially the hands-on experience and the visual output of robotics was perceived more intriguing in comparison to traditional lectures. [29]

Caldwell et al. [5] used robotics in an introductory class to computer science. The course consisted of the basics of programming, hardware, and computer architecture. They had experienced that many of the students found the course boring, which essentially affected their learning. Robotics was used to engage students in an interesting way. Students built and programmed sumo robots with Parallax robotics kits. As a result, the atmosphere of the course changed from boring to engaging. [5]

Also Garcia et al. [10] used robotics in engaging students in a computer science course. Students built robots which competed against a mouse finding a cheese from the maze. Results were similar to Caldwell et al. [5] results: students' engagement was positively affected by the use of robots. [10]

Contextualizing is one of the rising topics in education. Especially there is a rising demand for computing teachers to bring real-life context into teaching [21]. Real-life context and hands-on experience seems to provide good learning results [29]. Educational robotics is one way to answer into this demand.

According to Salgian et al. [23] there are few reported efforts to combine engineering with humanities and social sciences. Some interdisciplinary teaching studies can be found with literature review. For example, Brunvand and Stout [4] describe a cross-disciplinary course combining art and computing.

There are many existing toolkits and framework for educational robotics available today. A good example is Lego Mindstorms robotics sets. Mindstorms is one of the commonly used robotics sets in schools and children's technology clubs. It is also used in SciKids' clubs and will be discussed with more detail in Section 3. Mindstorms includes a set of building parts with essential electronics to build robots. It also ships with programming environment for PC and Mac. A robot built with Lego Mindstorms EV3 set is presented in Figure 2. [26]

The focus of this thesis is in educational robotics, but many of the principles are also usable in other fields. This applies for example to simply driving motors and reading sensors in any robotics application. In Section 3, I will present projects in which educational robotics has been implemented in practice.


Figure 2: Lego Mindstorms EV3 - educational robotics set

## 2.2 Storytelling and theatre

Storytelling is an ancient method of making the sense of the world and to pass knowledge to the next generations. Often, parents use storytelling to make the world more understandable for their children by telling stories using metaphors, which are easy to perceive. [6]

Our cultural heritage is full of stories and legends. For example, Finnish national epic Kalevala [15] was collected and compiled by Elias Lönnrot by interviewing old oral folklore singers. Kalevala tries to explain world and its existence, at the same time trying to deliver lessons. Storytelling is a powerful way to deliver information and cultural heritage.

Digital storytelling is basically the same as the traditional storytelling, but its enhanced with digital tools. In digital storytelling, digital media is used to express, store and share stories. Digital stories can be presented for example using video, animation, photographs, sounds, music, text and narrative voice. Basically, digital storytelling opens constantly new ways for presenting stories with new digital technologies. [6]

Digital storytelling is an efficient tool in education. It can make variation to traditional education, be used as examples, personalize learning experience, make learning more compelling and engaging, encourage students to express their own ideas, and bond classes through shared experiences. Most often, digital storytelling is associated with arts and humanities, but it could be also effective in teaching natural sciences. [22]

Theatre is a form of storytelling usually including actors playing the parts of the stories to make the story more vivid. Without a story, or a message, there is no theatre. Digital storytelling can be added to theatre performance to make additional effect, for example, by including digital media in performance.

Theatre is an enactment of a play and tries to tell a story, or deliver a message, to its observers. The actors, the director, and the staff together build a performance which creates an imaginary world incorporating on the stage for the audience to enjoy it [14]. Figure 3 illustrates an ongoing theatre play called "Oliver!". As seen from the picture, the imaginary world is made as immerse as possible with costumes, staging, and actors' creative performances.

To complete a theatre play there are usually many parts to play. The playwright produces a story as script, which the theatre's staff will like to present to its audience. Actors will play the characters of this story on the stage, which will provide a physical context for different scenes of the story. The director will guide the actors and the staff through the performance and his / hers job is to interpret the playwright's script. There usually is also a need for background staff supporting the whole performance. All these different roles need to work together to succeed. [14]

Theatre can be used as an educational tool by developing performances, delivering stories, and sharing knowledge. By including digital storytelling, theatre can be used as more multidisciplinary platform combining natural sciences with arts and humanities.

## 2.3 Theatre robotics and robotic storytelling

Existing educational robotics sets offer great tools for teaching natural sciences. There have been many studies and reported experiments around teaching technical subjects with robotics, but there are only a few cases where robotics and, for example, arts are

Figure 3: Ongoing theatre performance [1]

combined [23]. That indicates that there is a need for a more suitable framework for multidisciplinary educational robotics. Already existing robotics sets could be used for this kind of activity, but there are some issues, described in Chapter 3.6, which support the development of a new platform.

Jormanainen et al. [28] devised a concept of *theatre robotics*, to create this kind of needed multidisciplinary framework. The idea of theatre robotics is to give the students tools to design and implement a theatre performance with robotics set. To be able to successfully create a play with the theatre robotics framework, students need to practice several school subjects. A theatre robotics project can be described as a process which has several phases. These different phases will force students to focus on different subjects to successfully move to the next one. [28]

First, students need to select or create a story to be implemented with the framework. This phase might include school subjects like history or religion, which provide good stories on their own. These stories can be found from books, Internet, or from spoken heritage. Of course stories can be created by the students if it seems suitable for the cause. In the second phase information given by the selected story needs to be collected and processed. Students need to understand the story and form a mental model of it so

[1]Source: http://www.theguardian.com/stage/theatreblog/2009/apr/17/bbc-musicals-west-end, accessed February 5, 2015

they can implement it with the framework. [28]

The third phase is designing and building the actual robots. This includes arts and design, IT, engineering, handicrafts, and mathematics and physics. In the fourth phase, the created robots need to be programmed / scripted so that they will carry out the actual play. This phase is heavily computer science oriented and will practice programming and logical skills. [28]

The last phase is combining everything into one theatre performance. To create a complete play, students need to understand arts, social skills, and technology. By simply building and programming of the robots in the third and the fourth phase is not enough to complete a play. Robots' behaviour and design need to be honed until everything fits together. [28]

The theatre robotics concept was tested out in an experiment which is described in Section 3.4. The experiment worked as a proof of the concept and was carried out with already existing technology. In the next chapter, this experiment with other related studies will be presented. These experiences will be used as a basis for implementing the theatre robotics framework.

Theatre robotics is clearly a form of storytelling, using elements from digital media. Nevertheless, I claim that theatre robotics would not directly fall under the concept of digital storytelling. I would like to propose new term ***robotic storytelling*** to describe storytelling with robotic systems. As digital storytelling is traditional storytelling enhanced with digital media [6], robotic storytelling is the enhancement of digital storytelling with robotics systems. In addition to digital media, such as video and sound, robotic storytelling makes use of actuators and sensors, allowing computers to interact with the physical world.

In *robotic storytelling*, sensory data can be used to form models of the world into robot's memory. This model can then be used in decision making and reacting to the situation. Motors and other actuators will allow robots to affect and interact with the physical world in many ways. Robotic storytelling adds a new layer to digital storytelling.

# 3 Context: SciKids'

SciKids' is a collaborative research laboratory, which original idea was designed in University of Joensuu, 2001. At that time, SciKids' was called Kids' Club [8]. These days, Kids' Club is organized by Joensuu Science Society with the name SciKids' [12].

SciKids' is an afternoon club where children have an opportunity to get creative and play around with technology [9]. The idea is to provide different activities around technology, which at the same time educates and interests club members. During the recent years, focus has been in robotics, especially in building robots with Lego Mindstorms educational robotics sets.

I have been working for several years as an instructor in SciKids' technology clubs. We have been playing around with different robotics projects and came up with various solutions for different challenges. For example, we have successfully built soccer, rescue, and dance robots that have competed in RoboCupJunior championship in Finland for several times. All these robots have their own special characteristics to take on count when developing working solutions. Tutoring has given me some viewpoint for educational robotics and I have used that for designing the platform for theatre robotics.

Interest in club activities is maintained by setting different goals for the members. For example, a common goal has been taking part in the national RoboCupJunior competition of Finland. In RoboCupJunior, participants need to build robots for the tournament where they want to compete in.

For the researchers, SciKids offers a platform for studies and different experiments [12]. For example, in Section 3.4 an experiment towards theatre robotics is introduced. SciKids' environment played a critical part for my study, by providing crucial insights of the field, technology, and the target environment.

In the next subsections, I will present SciKids' clubs where I have been taking a part as an instructor. SciKids' experiences were documented by taking notes and pictures during club activities.

## 3.1 SciKids' Joensuu

SciKids' Joensuu, previously named Kids' Club, is the original children's club run in Joensuu. Club has been running since the year 2001, providing different activities around a constant theme of technology [8]. Club is mostly run by students from natural sciences, like computer science and physics. I have been tutoring SciKids' Joensuu since the year 2011. SciKids' season starts in autumn and lasts over the winter to the next spring. Club members gather once in two weeks at the university premisses to build and experiment with technology.

During the past few years, SciKids' has been focusing on robotics with Lego's Mindstorm robotics set. Lego Mindstorms is an easy way to get into the world of robotics for many children, because most of them are already familiar with Lego blocks. With Mindstorms, as with any Legos, it is fast and easy to build different, even complex, structures. Lego Mindstorms includes programmable brick, where different motors and sensors can be attached. Mindstorms set also includes a free graphical programming environment to create programs for your robots.

Children, who are interested in competing in robotics, are given opportunity to participate into the annual national RoboCupJunior competition of Finland, which determines each year's champions in different tournaments. Competition challenges are soccer, dance and rescue. Dance tournament shares elements with theatre robotics. Figure 4 presents a pair of soccer robots built for RoboCup Junior in 2013.

The first weeks of each year's club start with getting familiar with the robotics set and programming environment. After reaching appropriate skill level, children start preparing for challenges in which they are interested in. According to experiences from SciKids' Joensuu, it is clear that with Lego Mindstorms set it is quite trivial for children to come up with different creative physical structures and solutions for problems. Programming seems to be the most problematic part of robot construction.

Lego's programming environment, illustrated in Figure 5, makes it straightforward to create simple programs, but when there is a need for more complex ones, the environment might become hard to manage. All commands are presented with graphical blocks that are assembled into logical order to form algorithms for the robots. If the size of the program grows even to ten blocks with some logic structures, the program starts to take a considerable amount of space with an average laptop screen. This leads into a need for scrolling when the program needs to be observed, thus making it harder to understand the program.

In addition, Lego has proved to be durable in the hands of children. From time to time blocks and electronics have rough times by falling from the table and rolling in the children's feet. One Lego Mindstorms set seems to endure several years of active use.

For the theatre robotics development SciKids' Joensuu offers three important observations. First, Legos are easy to build with. Secondly, creating small programs with Lego's graphical programming environment is easy. Lastly, Lego Mindstroms sets are durable, which is important in club activities of SciKids' kind.

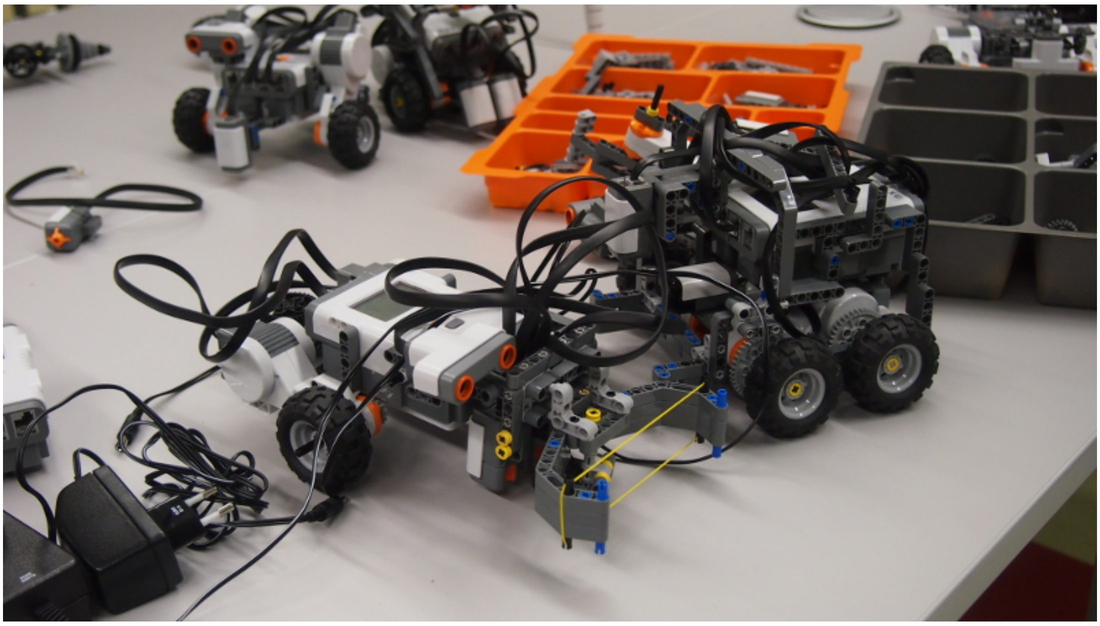

Figure 4: Soccer robots built in SciKids' Joensuu

## 3.2 SciKids' Kampala

During spring 2013, I travelled to Kampala, Uganda to spend three months in Makerere University Business School (MUBS) as an exchange student. My aim was to establish a local technology club for children, by first teaching a group of local students the essentials of educational robotics.

In Uganda, I was able to get a group of local bachelor students of business computing, who were interested in robotics. The group consisted of eight students in total. The

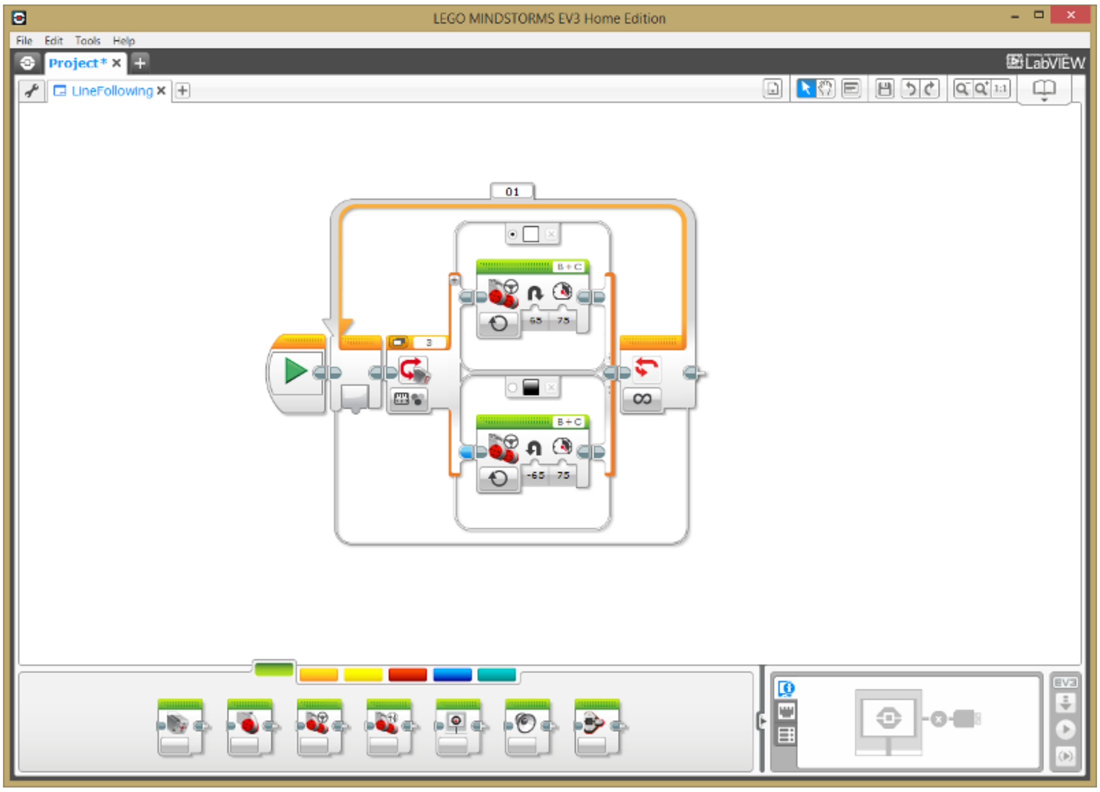


Figure 5: Lego Mindstorms EV3 programming environment

group was formed by my supervisor Dr. Bada by querying interests towards learning robotics within the students. Students had good computer skills, with some knowledge about programming, but they had no experience in building robots.

As a teaching tool, I used a set of bare electronics, described in list below, and Arduino prototyping boards. Arduino boards are easy to learn, low-cost technology, and because of that very usable as an educational tool. Arduino boards are introduced in more detail in Chapter 4.5.1.

Electronics used for getting familiar with robotics:

- A couple of Arduino UNO boards
- A set of resistors with various resistances
- A set of LEDs (Ligh Emmitting Diodes) with various colors
- Breadboards for creating connections without soldering
- Jumper wires for connecting components

- DC motors
- Servo motors
- Ultrasonic sensor, can be used for detecting distance
- Various other basic components such as capacitors, light dependant resistors, etc

Great amount of time was used to get familiar with the basic electronics. Studying was started with simple connections, like how to connect a LED to a resistor and a battery to form a closed circuit to light up the LED. From this, we continued to learn by adding slowly more components to form more and more complicated circuits.

By including Arduino in our circuit designs, it was possible to extend the functionalities of circuits and dive into the world of programming. We studied Arduino's capabilities by creating different circuits around it and controlling the circuits with different programs. The first test was Arduino's "Blink"-sketch, which blinks one led on and off as long as Arduino is powered up. In Arduino environment programs made for Arduino boards are called sketches.

When students gained enough knowledge about our electronics set, it was time to apply this knowledge in practice. We started to build an example robot by including Arduino and electronics in recycled toy parts. As a result, they were able to build a functional robot which was capable of roaming around in a room at the same time avoiding objects.

At the end of my stay in Uganda, we were also able to gather a group of children, interested in robotics, from one of the local schools. There were around ten children, ages between eight and twelve years. With three local students, who were taking my lectures, we were able to visit the school and give basic lessons about robotics. In Figure 6, one of the kids is building a battery powered circuit with multiple LEDs and switches.

The problems encountered with bachelor students were mostly about understanding circuits. Debugging the circuits and for example solving the most common problem, loose connections somewhere in the circuit, took a lot of time. However in the end, students got faster in mechanical tasks of this kind and developing circuits got easier. Programming with Arduino's environment was quite easy to learn because of basic knowledge about programming. The students also stated that understanding the code was easier because you could see the results of the code immediately in circuits operation.

The most important experiences for the theatre robotics were following. Getting familiar with bare electronics takes more time than simple Lego blocks. Programming seems to be more complicated, but at the same time allows creating more complex programs. Arduino boards are easy to attach to different platforms allowing the use of any available tools and materials.

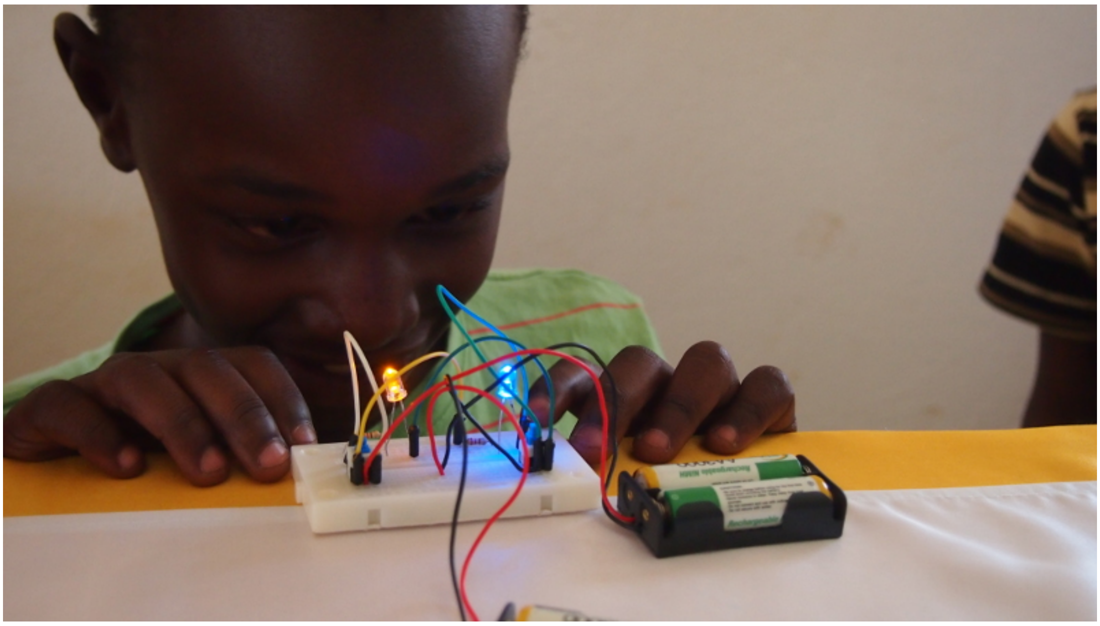

Figure 6: Building circuits in SciKids' Kampala

## 3.3 SciKids' Experts Joensuu

In the autumn of 2013, a new SciKids' group was established to extend learning opportunities in SciKids' Joensuu. Group was named SciKids' Experts due to the more challenging nature of the club activities. The aim of the club was to move from easy and familiar Lego robotics to more advanced electronics and programming.

A new group was formed by children who had been participating in SciKids' activities earlier and already had some experience from robotics. We were able to get a group of six excited participants, who were all about 10 years old. The club was held once in two weeks, two hours per session. SciKids' Experts was instructed by me and one postgraduate student from physics department.

Technology used in this club was similar to one used in Ugandan context described in the previous section. Arduino boards were used as central components and a large set of electronics were used to implement different solutions around them. The most essential parts are listed below:

- Arduino Mega 2560 boards
- Motor control shields for Arduino boards
- Resistors
- LEDs
- Breadboards
- Jumper wires
- DC motors
- Wheels and rubber tires fitting the used motors
- Servo motors
- Ultrasonic sensor, used for detecting distance
- Light detectors
- Metal frames for building robots

Our aim was to create a new competition to SciFest 2014, an annual science festival held in Joensuu. The idea was to build firefighter robots which could dive into a labyrinth containing randomly placed fires, in this case candles. All the candles should be found and extinguished. In overall, this proved to be quite a challenging job consisting of two main tasks: navigating through the labyrinth, and detecting and extinguishing fires. Figure 7 illustrates SciKids' Experts' table at SciFest 2014.

The physical structures of robots were following in all of the cases: robots had two or four motors with wheels, there were an Arduino board with a motor controller shield to handle the logic, one or several ultrasonic sensors to detect distances to make it possible to sense the labyrinth, and a light sensor to detect the fires. Building up the robots into a point where they were stable enough to survive inside the labyrinth was a challenging task and took approximately half of our time.

Even more challenging part was the programming of the robots. Arduino boards are programmed with C language with a set of libraries developed by Arduino developers. Arduino's development environment is described in more detail in Chapter 4.5.1.

Overall, it is relatively easy to start programming with Arduino because there are a lot of ready made examples. Even though learning programming is not an easy task [11, 13] and we were fast to notice this while getting familiar with Arduino programming. In the end, we decided to provide an example program for our club members. Then they needed to alter the program to suit their own robot's attributes. Even with the ready-made algorithm, most club members were not able to get their robot work during our club sessions.

SciKids' Experts provided important observations about programming and building robots with Arduino for the theatre robotics development. As with SciKids' Kampala, programming seemed to be the most problematic and hardest subject. Also, to built stable bases for robots takes time and effort more than with Lego blocks. However, Arduino allows certainly more freedom into building and programming.

Figure 7: SciKids Experts' working with firefighting robots at the Scifest 2014

### 3.4 SciKids' Theatre robotics Joensuu

In order to test the concept of theatre robotics, a new SciKids' club was organised during the spring 2014. The aim was to test existing robotics technology to implement a theatre performance with self-built and programmed robots.

Participants for the club were children who had previously participated in SciKids' activities. We were able to get eight interested participants, with age range from 8 to 12 years. The experiment was conducted within four months, from January to April, by meeting with participants once a week. Group was supervised by the same instructors as SciKids' Experts. Our plan was to build a theatre play for the annual Scifest science festival held in Joensuu, Finland.

Lego Mindstorms EV3 robotics sets were used in the project. All children attending to our theatre robotics group had some experience with older Lego Mindstorms NXT sets. The EV3 package contains a fair amount of building parts to form a body for a robot and different motors and sensors which can be used to sense and alter the environment around robots.

Process started by club members deciding a proper play to implement. After debating, they chose Red Riding Hood, a classic story by Grimms brothers. The fairytale is well known in Finland and club members thought that it would be interesting and they were familiar enough with it. The story has only four characters (the girl, the grandmother, the hunter, and the wolf), thus it was suitable for our resources.

Club members created the robots concurrently with designing the stage for the play. The robot building was fluent and any significant problems were not presented. The stage was created with cardboard and cloth, containing a forest and grandmother's cabin. Robots were decorated with different pieces of cloth and suitable Lego extensions.

After having building in a phase where robots were able to move around on the stage, the programming phase was initiated. Moving the robots freely around the stage was a simple task for experienced club members, but the problems emerged from the need to accurately control the robots on the stage. The same commands do not always result in exactly the same amount of movement. The problem was solved by making the robots follow coloured lines drawn on the stage. Club members had struggled with solutions of this kind in the previous clubs, so the solution was implemented fast. This also made possible to trigger events with different colours, for example a dialogue in a specific spot while the characters were travelling through the forest.

Even with this kind of well working solution, the programming part proved to be the most difficult task and it took so much time that we needed to scale down our play to contain only one scene. Selected scene was the one where Red Riding Hood comes to the grandmother's cabin where the wolf is waiting for her disguised as the grandmother.

As noticed also in other clubs, children built the robots relatively fast. Different solutions and structures were used innovatively. Similarly, programming seemed to be more difficult and took a lot of time.

Lego Mindstoms proved to be usable also for theatre plays. The set is durable and easy to start with, but there are also some aspects that support the need for a specific robotics set for theatre. First of all, accurate movements around the stage are crucial part of a theatre play and it is not trivial to implement with the Mindstorms set. Secondly, Lego Mindstorms set, especially the new EV3, is quite expensive. One set costs of 350-400€, so cost for four robots needed in many theatre plays does the total sum of 1400-1600€.

## 3.5 Lessons learnt

In this section, I will summarize the collected experiences from SciKids' technology clubs. The experiences are extracted from the notes and pictures collected during the club activities. This data will be used in the next section for analysing the essential aspects for the cause of the theatre robotics platform. Table 1 shows the most important observations with short descriptions.

It is clear that using Lego sets as tools for building robots is efficient because most of the children are used to creating structures with Legos. Legos have also proven to be durable and long lasting solution for SciKids' like environment. It is safe to assume that purchased Lego sets will last for several years. However, the price of 350-400€ for one Mindstorms set is high, especially when there is a need for several sets.

Arduino takes robotics to a more complex level. Building robots' structures gets more complicated and durability is not as good as with Legos. Creating proper circuits

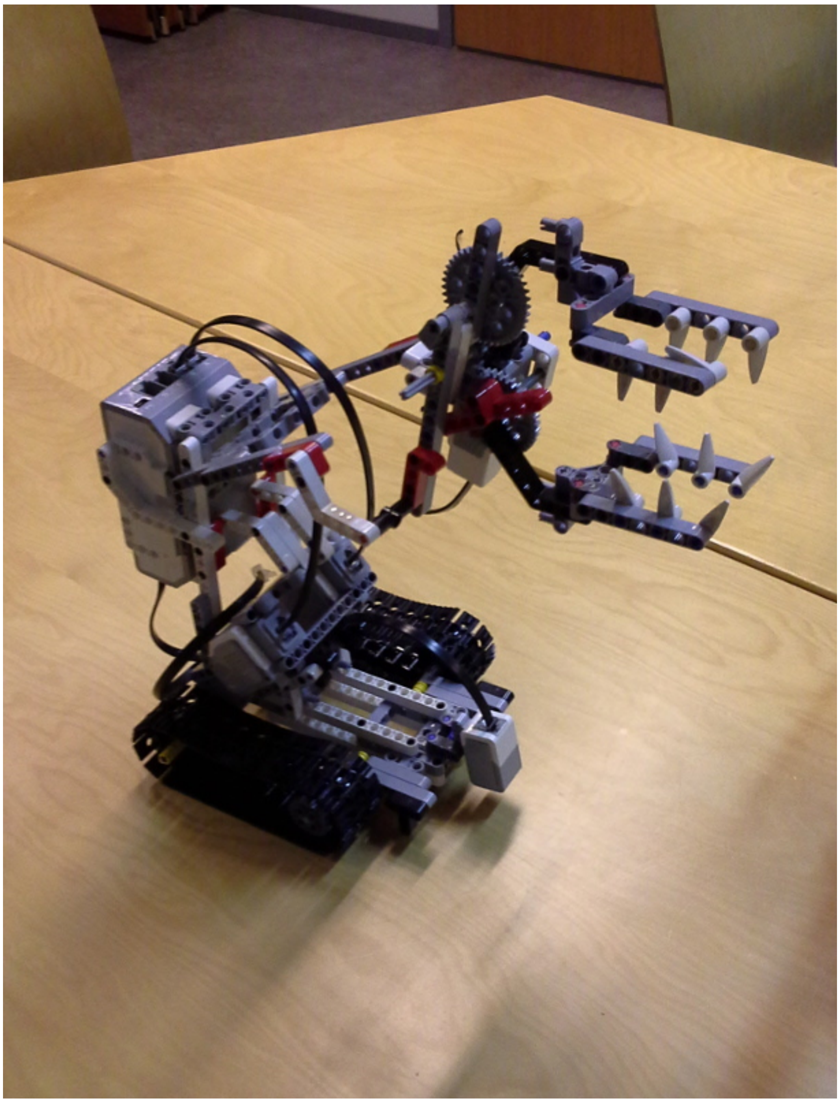

Figure 8: The skeleton of the wolf robot built by SciKids' Joensuu theatre group

is hard and takes more time in comparison to the Lego Mindstorms' way of simply plugging cables into correct ports. However, Arduino does not restrict building as much as Lego and is very cheap, even with additional components. With a price of one Lego Mindstorms set it is possible to get components for approximately four Arduino based robots.

Robots built with Mindstorms set are easy to program, as long as the size of the program is reasonable. With larger programs, Lego's environment becomes difficult to handle. Arduino's C based language is more complicated and harder to start play around with. After reaching enough knowledge, code is easier to structure and there are only a few restrictions, such as lack of threading.

## 3.6 Implications for theatre robotics development

In this section, I will describe what implications the above mentioned observations make for developing the theatre robotics platform.

Table 1: Central observations from SciKids' clubs

| *Observation* | *Description* |
|---|---|
| Building with Legos | Building is fast and solutions are achieved reliably. Legos are also very durable. |
| Programming with Lego | Easy to start with, but complicated and hard to handle larger programs. |
| Lego's price | A price tag of one Lego Mindstorms set is quite high, approximately 350€. With one Mindstorms set it is possible to build one robot. |
| Building with Arduino | Creating circuits might be complicated at the beginning, but challenges to use any means when building the form of a robot. |
| Programming with Arduino | Harder to start and understand, but very versatile and powerful. |
| Arduino's price | Arduino costs about 20€, but needs other components in order for building a robot around it. When adding crucial components like motors, sensors, cables, and frame the estimation for one robotics set is from 50 to 100€. |
| Accurate moving | Accurately moving a robot to a specific spot is not easy with Lego Mindstorms. |

There are many factors that make Lego Mindstorms set very suitable for building educational robots. Children are familiar with Legos and even if not, they learn it quickly. Legos are also durable and combining small programs with graphical programming environment is fluent, fast, and easy. For many schools and small organizations, Mindstorms' price can be too high. Also, programs with large logic are hard to manage with Lego's software.

One problem in using Legos for theatre robotics is robots' accurate moving on the stage. There is a clear need for implementing an indoor positioning system which can keep track of the robots' locations. Implementing this to suit Mindstorms does not seem to be possible. Arduino's open environment offers a better support for building extensions of this kind.

Observations from previous SciKids' activities implicate that there is a need to build a new platform for theatre robotics. The price of the platform will be a significant factor and because of that, Arduino is a good way to go. By using Arduino in robotics projects, there are problems which we would not come by with Legos, like more fragile parts and in overall harder building and programming phases. Arduino's open source nature provides a good base for developing the theatre robotics platform.

The next chapter will focus on defining design for the theatre robotics platform whose foundations are based on observations made in this chapter.

# 4 Platform Design

The following sections introduce a design for the theater robotics platform. First, in Chapter 4.1 the requirements for designing the platform are defined. Then, in the next section, an overview of the whole system is described. The rest of the sections describe the individual modules of the theatre robotics platform in more detail. Technical details are discussed with moderate technical orientation, without digging too deep, but still trying to highlight interesting aspects. Defining the technical details is not relevant in this phase of the designing process and are not in the scope of this thesis.

## 4.1 Requirements

It is crucial to successfully identify requirements for the designed artifact. Defining relevant requirements is a key to a successful implementation of a project. In this section, I will focus on finding and defining these requirements for the theatre robotics platform.

As described in Chapter 2.3, the aim in this thesis is to create a plan for implementing a theater robotics platform, which can be used as an educational tool for teaching various subjects. The goal can be reached by designing the platform in a way that a successful implementation of a theater robotics play requires knowledge and skills from several subjects.

Engineering, crafting, and physics can be taught with the building phase of the robots. The platform needs to include a suitable toolkit of components to build robots. The set of components should be comprehensive enough to make it possible to create robots and constructions of different kind. At the same time it is required to keep in the mind robustness and durability of the toolkit. Also handcrafting should be encouraged, so that users of the set would not stick only with the provided parts and equipment. As mentioned earlier, one crucial aspect is the price of the platform. The price should be affordable enough for small schools and communities.

Durability of the designed set need to be considered with care. As described in earlier chapters, Lego Mindstorms can be credited about this aspect and with bare electronics it is clearly a problem.

Computing is taught when designing software for a play. SciKids' clubs have shown that Lego Mindstorms' accessibility is good and the environment is intuitive with small programs. Then again, when programming Arduino, it is a huge step to get comfortable with C programming, but afterwards it is a way more powerful environment. Programming the behaviour for theatre robots should have Lego's accessibility and easiness, at the same time having C's powerfulness and functionalities.

One of the problems identified from SciKids' theatre robotics experiment with Lego Mindstorms was positioning robots within the stage. To make programming reasonable, implementing a system which can keep accurately track of robots is needed.

The actual design of the platform is presented in the following sections. Table 2 summarizes the main requirements for the implementation.

Table 2: Summary of defined requirements.

| *Requirement* | *Short description* |
|---|---|
| Comprehensibility | Platform should be comprehensive enough to support creating different structures and solutions. |
| Durability | Used physical parts should be durable because of the nature of target environments of kids' club kind. |
| Affordability | The price of the designed platform should be affordable. This should be taken into account when considering electronics and programming tools for the platform. |
| Accessibility | Accessibility of both programming and building is essential. Learning programming is generally recognized to be a hard task. The problem can be, at least partially, solved by making use of known good solutions. In building, accessibility needs to be considered in component design. |
| Positioning robots | For the sake of implementing successful theatre performance with robots, it is crucial to be able to position robots accurately on the stage. |

## 4.2 Overview

The theater robotics platform is designed in a modular manner. There are four main modules which communicate with each other to form a flexible platform for creating

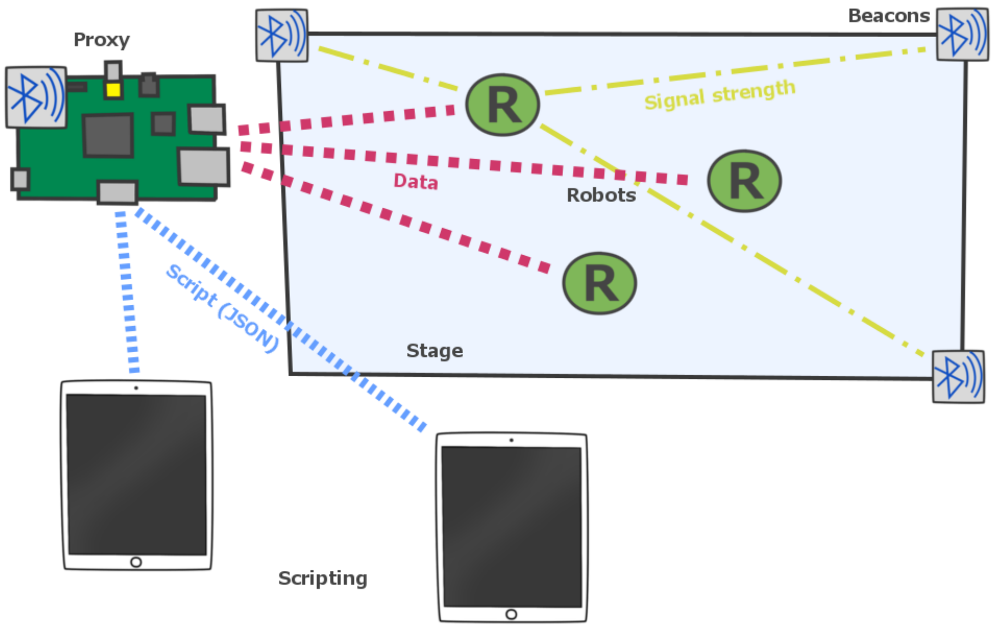


Figure 9: Architecture of the theatre robotics platform [28]

theatre plays. The modules are a scripting environment, a stage, a proxy and a robot actor toolkit. Figure 9 visualizes different modules and their relations.

Scripting environment is an IDE (Integrated Development Environment), which users can use to form the logic of the play. Scripting environment can be considered as programming environment, but perhaps more accurate definition would be scripting. This way, users will be assigned the role of a playwright when implementing the play.

Proxy is a gateway between scripting environment and robot actors. Proxy's role is to work as a director of the play. Proxy acquires a script of a play from the scripting environment and then directs actors' performance according to that script.

Robot actors are the workhorses of plays. They carry out the actual performance as the director - the proxy - wishes. Robots are obedient towards the director and in a way just dummies with capabilities to mechanical work. Robot actors are built by the users with a toolkit including essential components in the robots.

The stage module's role is to work as a stage in the play, but it should also implement the indoor positioning system for tracking robots' positions during the play. The stage will communicate with the robots and the proxy to provide data, which can be used to define positions for each of the robots.

The next sections describe the individual modules in detail. First, the scripting environment is described in Section 4.3 and the proxy in Section 4.4. Section 4.5 discusses the robot actor module, basically meaning the robot building toolkit. At last, the stage module is described in Section 4.6.

## 4.3 Scripting environment

To establish a successful theatre play, there is a need for some kind of manuscript to guide the director to lead the actors. In regular theatre, the manuscript can be written on paper and used as a tool by directors and actors. In the context of robotic storytelling creating a manuscript for a play means programming the behaviour of the robot actors.

Instead of programming, a more descriptive term would be scripting. That means rising the abstraction level higher, in comparison to programming languages, so plays could be scripted in a similar manner than at regular theatre. For example, there could be statements like "Robot A, move to Robot B and greet him", instead of "Robot A drive motor #1 and motor #2 for 3 seconds, then drive motor #2 for 2 seconds...". Of course, there might be also a need for these kind of low-level commands.

As reported on Chapter 3.6, programming is one of the main issues encountered in educational robotics. This needs to be taken into account when designing and defining requirements for the scripting environment.

### 4.3.1 Requirements

Chapter 3.6 describes the observations from SciKids' technology clubs that Lego Mindstorms' development environment has many suitable features, such as a nice graphical user interface and great accessibility. However, writing C code seemed to be more problematic and take more time. Also, to make it more portable, scripting environment should be developed as mobile application designed especially for tablets.

A mobile approach allows lowering the price tier, because there are plenty of low-cost tablets available. That leads also into a restriction of developing only for Android OS, because low-cost mobile devices mainly use it.

Writing scripts for a play with mobile devices does not seem reasonable and easy to use. Robots' interactions need to be presented in more abstract manner and Lego Mindstorms graphical approach (see Figure 5 in Chapter 3.1) seems to be the most suitable for a mobile environment. There should be a set of readymade blocks, like in the Mindstorms environment, which users could use. Also possibility to construct self-defined blocks should be implemented. The programming, or scripting, is done by arranging the suitable blocks into a logical order by drag-and-drop mechanism.

There are three possible approaches for programming the robots: procedural (1), behavior-driven (2), and audience driven(3). Procedural programming is done in a similar manner than with Lego Mindstorms or Arduino. Every small action is defined by the programmer. In the behavior-driven approach, user could define robot's behaviour with other robots, in other words, what should a robot do in different situations. In this case, the programmer cannot actually know what will happen when the play is executed because rules can be defined at a more abstract level. The audience driven approach allows audience to affect the play. Members of the audience could have devices with a special interface capable of triggering events or changing some of the attributes in the play.

The advantages of the procedural approach are in the controllability of the play. When users define every and each of the actions beforehand, the play should always execute in a similar manner and the end result should be known. This allows the tweaking of the play until the result is exactly as desired. Then again, this approach will need more work from the playwrights.

The behavior-driven approach is more complex. Basically, when the behaviour is simply defined by stating rules for different events, it is impossible to accurately predict the result. For example, changing the starting points of the robots will change the outcome of the play. On the other hand, this approach would provide somewhat a more lightweight way for creating the script, when users do not need to write down actions in chronological order.

The audience driven approach, adding a possibility for the audience to interact or interfere with the play, could be appended with any of the previous two approaches. This could result in drastic changes in the ongoing play. The most difficult part in this approach is the development. Capability to interact with a play on the fly should be designed very carefully, so it can't ruin the system or the play.

All of these approaches are interesting and should be included in the scripting environment. The main approach would be the behavior-driven approach allowing high abstraction level. Capability to create actions also with the procedural approach is necessary. This will make the environment more powerful and enhance user's possibilities. In addition, the audience driven approach should be implemented, but it does not directly create the play and thus should be considered as an extra feature.

## 4.4 Proxy

One solution to control the actor robots is to connect scripting environment directly with the robots and handle controlling by the tablet. In this case, the tablet should maintain multiple connections if there are several robots at the stage. This will lead into draining of the battery and force the tablet to be within the Bluetooth range of the robots all the time. More reasonable solution would be to implement proxy, which will communicate with the scripting environment and then handle controlling the robots. In other words, proxy would play the role of the director of the play.

Proxy receives scripts from the tablet computer via Bluetooth connection. These user-made scripts are parsed and proxy will create a state machine presentation of the play. When the user wants to execute the play, the proxy connects with the actor robots and commands them to act the representation of the model of the play. Proxy observes robots states and gives commands in small tasks which robots are capable of executing by themselves. Examples of communication protocols are presented in Section 4.4.1.

The proxy has multiple tasks to handle. It needs to communicate with the scripting environment and the robots, and handle the execution of the play. Hence, there is a need to have a powerful but yet affordable computer as proxy. This also guarantees the extensibility of the system. Robotics storytelling extends the digital storytelling and use of powerful enough proxy allows digital media, such as sound and video, to be added to a play. The selected Raspberry Pi computer will be introduced in Section 4.4.2.

```
{
    "id": 0,
    "robot_id": 1,
    "actions": [
        {
            "id": 1,
            "cmd":"move_to",
            "target":[
                {"id":5, "type":"robot"}
            ]
        }, {
            "id": 2,
            "cmd":"greet",
            "target":[
                {"id":5, "type":"robot"}
            ]
        }
     ]
}
```

Figure 10: An example of possible JSON delivered commands

### 4.4.1 Communication

Communication between the proxy, the robots, and the tablets should be preferably lightweight. Especially when the proxy needs to communicate with the actor robots, the used protocol should be considered so that the robot's side is capable of handling it.

One possible solution is to use JSON (JavaScript Object Notation), to deliver messages between the modules. JSON is lightweight, easy to parse, and even readable for humans. Readability is important for development and debugging. JSON would be a suitable option for theatre robotics purposes. A sample of possible message is presented in Figure 10. [7]

Another interesting possibility could be using G-code, which is used in 3D-printers and CNC-machines. G-code is very lightweight and provides protocol for moving in XYZ-coordinate system. Using G-code could be usable especially between the proxy and the robots. A sample of G-code is presented in Figure 11. [20]

```
G1 X50 Y25.3 E22.4 ;move X to 50mm and Y to 25,3mm
                   ;while extruding 22,4mm of filament
```

Figure 11: A piece of G-code for commanding RepRap 3D-printer [20]

### 4.4.2 Raspberry Pi

Raspberry Pi is a credit card sized computer. It has a lot of processing power with 700MHz processor and 512Mb memory (model B). Raspberry Pi has ethernet, audio, and HDMI connections with multiple USB ports. SD card is needed to contain an operating system. Raspberry Pi can be powered with +5V power supply through micro USB connector. [27]

Raspberry Pi has not built in Bluetooth connection, thus a Bluetooth dongle is needed to connect with scripting environment and robots. In overall, Raspberry Pi is a cost efficient solution as proxy with an approximate price of 50 €. [27]

Raspberry Pi also has an active community, which makes the searching of information easy. This allows fast product development and helps problem solving. [27]

In comparison, there are also similar products like BeagleBone. BeagleBone has somewhat more processing capability with 1GHz processor (BeagleBone Black), but has a higher price [25]. In this case BeagleBone's higher price is not reasonable with the gain, because Raspberry Pi is clearly powerful enough.

With the Raspberry, it would be easy to add, for example, screens and speakers to the stage to enhance the performance with digital media. The performance's could greatly benefit from music tracks, narrative, videos, and animations.

## 4.5 Robot building kit

Robot actors are the workhorses of the theatre robotics platform and are in the most visible part in plays. The robot building toolkit needs to include all necessary parts to implement actor robots. The toolkit needs building blocks for the robot building, electronics for controlling the robots, and actuators such as motors for moving the robots. Also, sensors are needed to acquire data from the world surrounding the robots.

The theatre robot toolkit also needs firmware for the built robots. The firmware should be pre-installed to the used microcontrollers, or at least available for the installation. Firmware needs to make sure that communication between with the proxy is done properly.

In the next sections, these different components are presented and discussed. The selection of the components is done by criteria defined in Chapter 3.6.

### 4.5.1 Arduino

An autonomous robot needs a computational unit to perform actions. Most suitable units for use cases of this kind are small microcontrollers. A microcontroller is a small processor unit containing direct inputs and outputs that can be used to add sensors and actuators to sense and affect the real world. Actuators are usually motors and other equipment that can somehow "touch" the world.

When robot's physical form has been built so that it is capable of doing what it is designed for, it is time to add logic that adds actual functionality to it. This can be done with microcontrollers by programming the controller with an appropriate programming language. Arduino is a product of an open source project that includes a microcontroller on an easy to use prototyping board, resulting in a good platform for robotics projects of this kind. Figure 12 shows a commonly used Arduino UNO board. [2]

Arduino board is programmed with C-based programming language and Arduino project has an IDE (Integrated Development Environment, Figure 13) freely available for programming. The Arduino IDE is capable of coloring the syntax, compiling the code, and uploading it to the connected Arduino board. [2]

Figure 14 illustrates a small example program for Arduino. This code can be used to read messages from Arduino board's serial connection, for example, directly from a computer or Bluetooth connection. The setup()-function gets executed once when the board is powered, and in the example a serial connection is established. Setup() is run only once when Arduino is powered up. After the setup(), the execution of the program automatically moves to the loop()-function, where messages are read and written back to the serial connection. Arduino keeps executing the loop()-function until the board

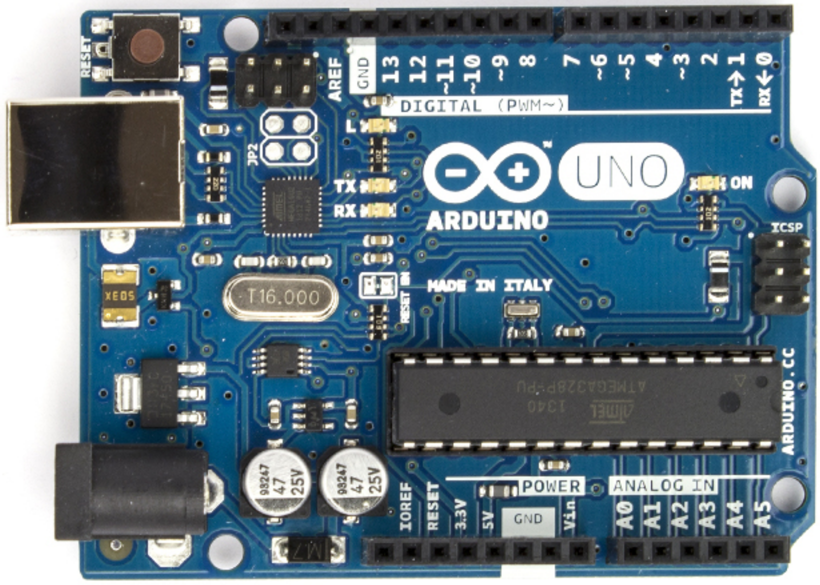


Figure 12: Arduino Uno [2]

is powered off. The example in Figure 14 is for illustrating an Arduino program, but it can be used as a base for writing the firmware for the theatre robots, because Bluetooth communication is handled with a similar manner.

### 4.5.2 Other electronics

An Arduino board alone is not enough to make a robot function. There is a need for various electronics components to realize movement, sensing, and communication (Figure 15). The most essential components for theatre robotics toolkit are listed below and then they are described in more detail.

- Breadboards
- Cables and jumper wires
- Resistors
- Capacitors

[2] Source: http://arduino.cc/en/Main/ArduinoBoardUno, accessed December 28, 2014

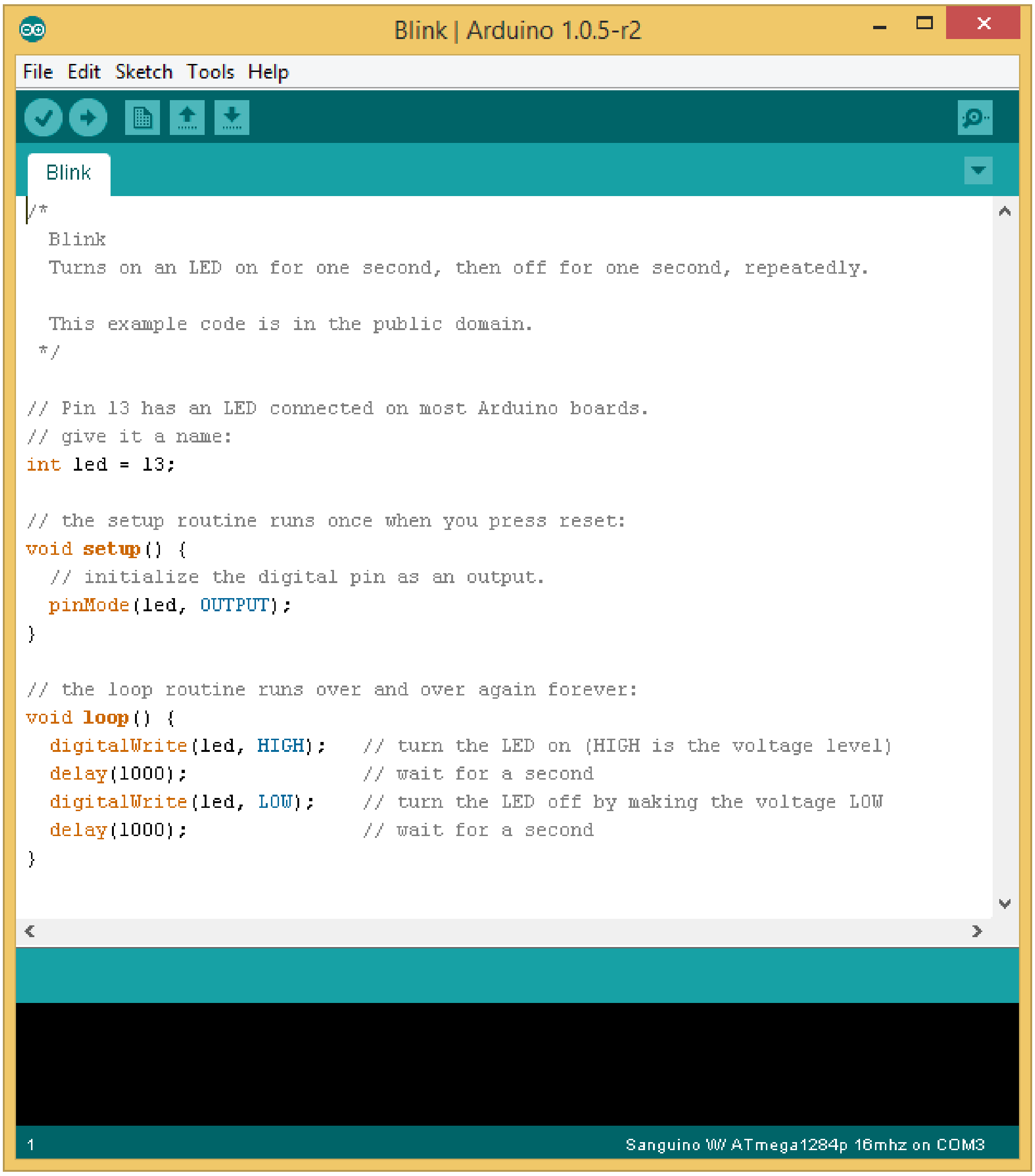


Figure 13: Arduino IDE with ready provided Blink-example

```
String data;

void setup() {
    Serial.begin(9600);
    Serial.println("Waiting for messages");
}

void loop() {
    while (Serial.available() > 0) {
        char received = Serial.read();
        data += received;

        if (received == '\n') {
            Serial.print("Received: ");
            Serial.print(data);

            data = "";
        }
    }
}
```

Figure 14: Receiving data from serial with Arduino

- LEDs (Light Emitting Diode)
- Buttons and switches
- Motors - DC and Servo
- Motor controller shield for Arduino
- Sensors - Ultrasonic, color, light, gyro, and acceleration

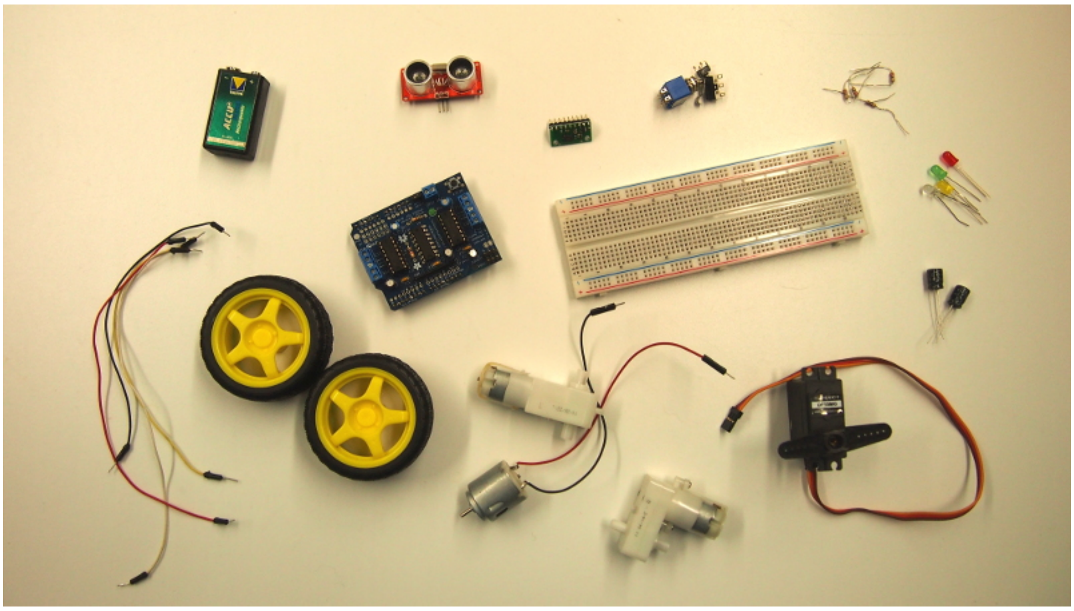

Figure 15: Essential electronics

Components are connected together to form complete circuits, which are capable of perform designed tasks. For example, a LED can be connected in series with a resistor and a battery, resulting emitting of light from the LED. Breadboards are tools for developing and testing circuits without a need for soldering components together. Jumper wires can be used with breadboards by simply plugging them into boards. Most of other components, such as LEDs and resistors can be used in a similar way.

LEDs can be used to test connections or to signal, for example, a state of a robot. Resistors are needed between components to ensure that current stays at a desired level. Capacitors are used for example to filter the current and prevent noise that can cause misbehavior in other components, such as motors. Buttons and switches can be used to give inputs to the program by affecting the flow of the power on the board.

Motors are needed to make robots move. DC motors (Direct Current) are simple to use for moving robots around. DC motors have two input ports and the motor starts turning when other port has a positive voltage and another port is used for grounding. The direction can be switched by changing the order of the ports. Also, changing the used voltage changes speed of DC motors. Two DC motors attached to wheels and tires enables the easily controllable movability of a robot. Figure 16 presents an Arduino code implementing functions for basic movements with two DC motors.

Servos are geared DC motors with an integrated control circuit and a potentiometer for detecting the turning angle of the motor. Most servos operate within a pre-defined angle, usually between 0 and 180 degrees. Servos have three wires: power, ground, and signal. Small servo motors usually operate with 5V power. Servo's control circuit automatically turns the motor to the position, which is received from the signal wire. Servo motors are great for turning the motor accurately to a desired position. This is useful when designing, for example, limbs for robots. To use servo motors and DC motors with Arduino, it is highly recommended to use a motor control shield. The motor control shield protects Arduino and it prevents the attached motors from draining too much power through the Arduino board.

Different sensors are used to sensing the world around the robot. Data provided by sensors can be used to form the logic of the program. For example, an ultrasonic sensor can be used to measure distances between a robot and objects nearby. Gyro and acceleration sensors can be used to determine the direction of the robot. This data is especially useful for a robot's navigation.

### 4.5.3 Building blocks

To build a robot, some physical parts are always needed. The theatre robot toolkit should include a set of multi-purpose building blocks. The easiest way would be using some existing building sets, such as Legos [26] or Meccanos [17]. Arduino boards and bare electronics can be easily combined basically with any material. This makes it possible to use, for example, recycled materials or new technologies, like 3D printing, to create robots.

3D printing is becoming more and more popular and cheaper. Many schools have their own 3D printers for example in physics classes. Figure 17 illustrates a 3D model of a

```
// Adafruit motor shield library:
// http://www.adafruit.com/product/81
#include <AFMotor.h>

AF_DCMotor ml(1); // left motor
AF_DCMotor mr(2); // right motor

void setup() {
    ml.setSpeed(200);
    mr.setSpeed(200);
    ml.run(RELEASE);
    mr.run(RELEASE);
}
void loop() {
    //drive a square
    driveFwd();
    delay(1000);
    turnRight();
    delay(500);
    stopMotors();
}

void driveFwd() {
    ml.run(FORWARD);
    mr.run(FORWARD);
}
void driveBwd() {
    ml.run(BACKWARD);
    mr.run(BACKWARD);
}
void turnRight() {
    ml.run(FORWARD);
    mr.run(BACKWARD);
}
void turnLeft() {
    ml.run(BACKWARD);
    mr.run(FORWARD);
}
void stopMotors() {
    ml.run(RELEASE);
    mr.run(RELEASE);
}
```

Figure 16: Controlling two dc motors with Arduino

servo mount for an ultrasonic sensor. Image was captured from www.thingiverse.com, which is Internet community where users share their self-made models.

This potential of 3D printing could be utilized in the theatre robotics project by creating a set of 3D-models of simple and usable blocks for the robot building. In this way, advanced users could also design their own parts and use their imagination to form creative bases for robots. This would reduce the number of default parts in the toolkit.

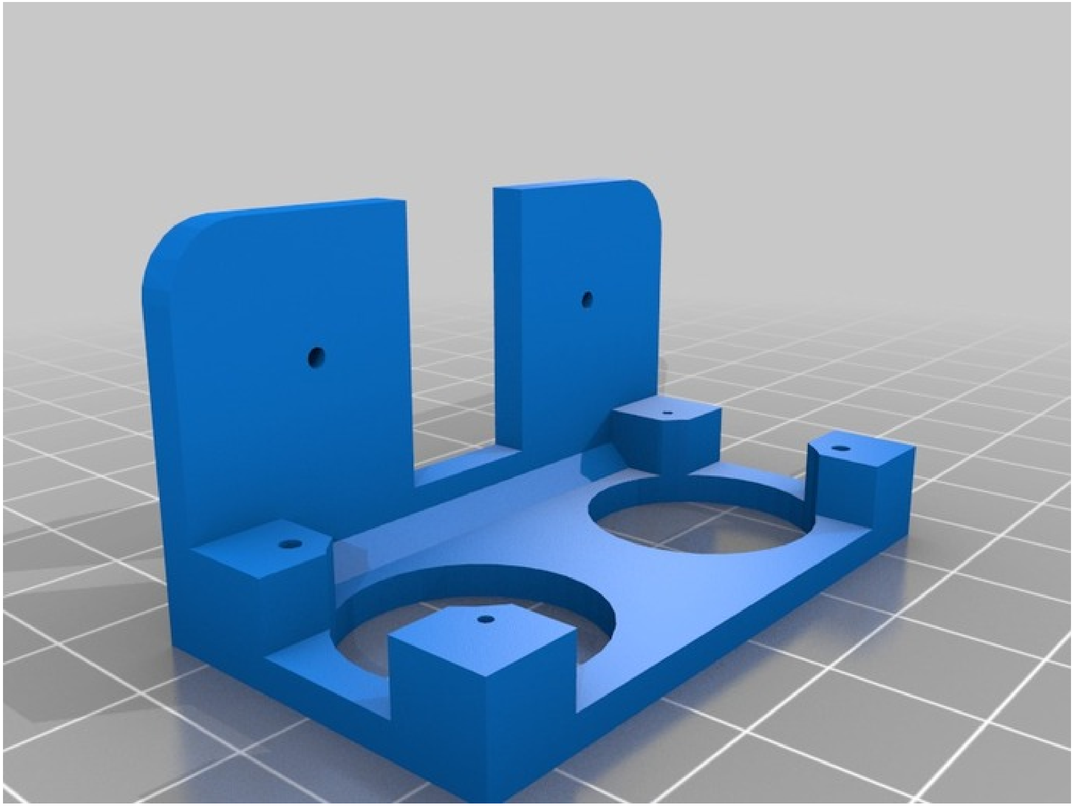

Figure 17: A 3D model of a servo mounting for an ultrasonic sensor. Created and shared by user f3rdys from http://www.thingiverse.com [4]

## 4.6 Stage

A stage is obviously an essential part of a theatre performance. The stage includes props to create visually impressive scheme and it is the environment where the actors operate.

In theatre robotics, the functions of the stage extend beyond the previously described features. As indicated in Chapter 3.6, one of the missing features of Lego Mindstorms

[4] Source: http://www.thingiverse.com/thing:100486, accessed February 20, 2015

is a possibility to detect the positions of robots inside a given space. Hence, in the theatre robotics environment, the stage should provide functionalities for that. This is especially important in the scripting environment where robots' functionalities are described with abstract commands such as "Find a cup of coffee and take a sip".

The stage of the theatre robotics platform should implement an indoor positioning system, which is capable of positioning robots accurately enough. There are a few appropriate techniques that could be used to realize the positioning system.

The first technique uses Bluetooth (BT) Low Energy (LE) signals to triangulate a position of robots. In theory, it works by setting BT beacons into the corners of a square stage and reading signal strengths between the beacons and robots. The position of the robot can be calculated from the signal strengths.

Study by Oksar [19] indicates that inaccuracy of BT triangulation might be a problem in the theatre robotics platform. In the study, Oksar used similar triangulation technique to detect a position with Root-Mean-Square-Error (RMSE) of 2.309 meters. This is not a suitable accuracy for the theatre robotics platform where the positions should be detected with the accuracy of tens of centimetres. On the other hand, there are commercial systems using more accurate BT positioning. This indicates that using this technique could be possible.

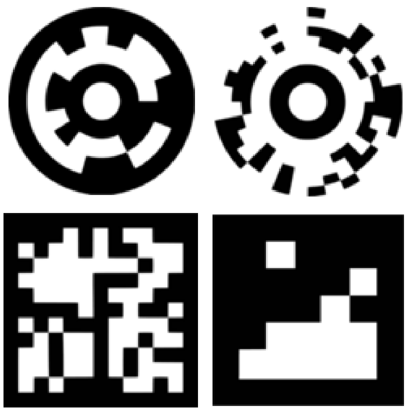

Figure 18: An example of fiducial markers [5]

Another way to achieve the desired result could be the use of a relatively simple machine vision system. The machine vision solution requires a method that can be used to identity the robots. For example, fiducial markers [24], illustrated in Figure 18, could be attached to the robots. Problems in this approach are related to visibility and the size

[5] Source: https://www.cl.cam.ac.uk/ acr31/cantag/tags.png, accessed February 23, 2015

of the markers. The props on the stage between theatre performances can vary much, possibly leading into situations where finding a robot inside the stage is impossible. Fiducials should be big enough so they can be detected reliably. This might affect the design and the looks of the robots. Another approach to identify a robot by using machine vision could be using infrared (IR) LEDs with a suitable camera. With a small LED on each of the robots, it could be possible to pinpoint robots on the stage with enough accuracy. The visibility problem remains, but the size of a led is small and does not interfere with the design of robots.

## 4.7 Summary of design

The previous sections discussed the possibilities to selecting hardware and approaches for the theatre robotics platform. The resulting architecture design for the platform is summarized as follows:

- Graphical scripting environment running in Android devices with the combination of behavior-driven and procedural approaches
- Small sized but powerful Raspberry Pi as proxy
- Affordable Arduino based robotics toolkit for building actor robots
- Stage with indoor positioning system (implementation technology needs more research)

In the next chapter, the design presented here will be evaluated.

# 5 Evaluation of the design

In this section, I will evaluate the platform design described in the previous chapter. Evaluation is based on criteria presented by Weiss et al. [30]. The following criteria have been used to evaluate the suitability of robotics sets for undergraduate education and it will be also used in this work.

- Flexibility and extensibility of the platform
- Quality / durability of the hardware
- Quality of the documentation and teaching resources
- Availability of demo code
- Level of activity of the developer community surrounding the platform
- Total cost of working systems (including hardware / SDK / peripherals)
- Availability / usability of SDKs for the platform
- Choice of programming languages

Table 3 summarizes the evaluation results. The results are discussed with more details below.

Table 3: Summarized evaluation.

| *Evaluation criteria* | *Designed toolkit* |
|---|---|
| Flexibility | Excellent |
| Extensibility | Excellent |
| Quality/durability | Fair |
| Quality of the documentation | Not able to evaluate yet |
| Availability of demo codes | Not able to evaluate yet |
| Activity of the developer community | Not able to evaluate yet |
| Cost | Good |
| Availability/usability of SDKs | Not able to evaluate yet |
| The choise of programming languages | Fair |

Flexibility of the toolkit is in generally excellent. The toolkit is designed for implementing theatre robotics, but it can be used for robotics projects of any kind. Designed scripting environment can be used to define robots behaviour in other context. Arduino also allows programming directly from the computer and using the scripting environment is not even necessary. In this way, the hardware of the toolkit can be applied in any kind of use.

Arduino board design also allows the system to be designed in very extensive manner. Adding own sensors or actuators is possible as long as connections between components are done properly.

The biggest issue in the theatre robotics hardware as discussed in this thesis is the quality and durability of the hardware. The toolkit is built with bare electronics and relatively fragile PCBs (Printed Circuit Boards). Arduino does not have any protective housing by default. This exposes the board for example to liquids or hits that will most likely destroy it. Also, the condition of bare electronics components will get weary in quite a short period of time because of the small size and fragile design of the components.

The cost of the toolkit is good. Estimation for the cost is calculated based on the approximate prices gathered for Table 4. [6]

Table 4: Approximate prices.

| *Component* | *Approximate price in €* |
|---|---|
| Arduino UNO | 5 |
| Arduino Mega 2560 | 10 |
| Raspberry Pi | 35 |
| Frame for a two wheeled robot | 10 |
| Bluetooth module | 3 |
| Other sensors (acceleration, ultrasonic) | 10 |
| Servo motor | 8 |
| DC-motor | 5 |
| Motor shield for Arduino | 5 |
| Collection of bare electronics (leds, etc) | 10 |

The price for one actor robot would be approximately 50€ and the cost of one stage module would be around 100€. In addition, to use the scripting environment, users would need one Android tablet or phone, where the lowest price tier is on between 50-100€. In total, the cost of theatre robotics kit with four actor robots would be approximately 400€. In comparison to Lego Mindstorms EV3 robotics set with cost

[6]Prices were obtained from www.ebay.com, February 2015.

of 350-400€, the designed theatre robotics toolkit is very affordable. It is possible to build only one robot with an EV3 set, but the proposed theatre robot set allows building four robots.

The set of usable programming languages with the designed set is quite small. If the scripting environment is used, then programming is done with the graphical manner designed for the purpose. Usage of Arduino boards allows the use of different approaches also in this case. Arduino can be programmed directly with C/C++, or even Assembly [2]. The communication protocol between actor robots and the proxy could be documented and released for the use for anyone, so developing new applications would be possible. Then, basically any language could be used to control the robots.

The following criteria were not in the focus of the evaluation presented in this phase of design and in the context of this thesis: quality of the documentation, availability of demo codes, activity of the developer community, and availability/usability of SDKs. However, all of these criteria are essential for a successful robotics toolkit and therefore they are still be discussed here.

Documentation of the toolkit is crucial in complex systems such as the theatre robotics platform. Assembly and user manual needs to be constructed for the users of the platform. Without a sufficient manual, starting to use the platform would most probably be impossible for the most of the targeted users. The documentation should include enough demo codes and practical demonstrations how the toolkit should be used. Also, SDKs (Software Development Kit) should be published with all the necessary protocols to allow interested developers and advanced users to get the most benefit from the platform.

Establishing an online community for the theatre robotics platform would be extremely useful. Community could provide crucial feedback for the development and, at the same time, support other users. Lively community would help the developers to manage a workload caused by customer care and help.

I am positive that using Arduino boards as central components in the theatre robotics platform is a well working solution. Arduino environment allows rapid development, alongside excellent extendibility and flexibility for the toolkit. Also, implementing the proxy module with a low cost Raspberry Pi computer seems a reasonable choice. Comparison to microcontrollers, Raspberry Pi has a vast amount of processing power and it will easily form a bridge between tablets and Arduino controlled robots.

The most unreliable part of the implementation plan is the indoor positioning system. For the usability of the system it would be highly recommended to get it work with suitable accuracy, but this is not crucial. It is certain that some sort of system with positioning capabilities is needed, but accuracy and performance requirements need more research. Previous studies indicate that positioning with Bluetooth is not very robust and accurate enough. In the machine vision approach for positioning, issues arise from shape restrictions for the robots. These aspects may lead to a need for having a stable enough system even without the indoor positioning system.

The whole theatre robotics platform needs to be well tested during the development process. The modular approach makes the system prone to faulty behavior, but this can be avoided with extensive enough testing. It is recommended that development builds will frequently be tested in real-world use scenarios, such as SciKids' technology clubs, for gaining authenticate feedback from children. In the evaluation, we are interested to see especially how the developed programming and controlling environment works, are the children able to set up a theatre play with the toolkit, and whether the platform lacks some crucial parts. The extensive evaluation of the implemented system is, however, out of the scope of this thesis.

# 6 Conclusion

Educational robotics is at the moment a trending tool in teaching. It offers an interesting and engaging environment for teaching various subjects. Robotics is mostly used in teaching natural sciences, especially computer science and engineering, but results from the experiences from these classes indicate that there would be a true benefit to apply robotics in more multidisciplinary manner. This thesis presents a design for the theatre robotics platform, which aims answering that need.

I am positive, that the findings of this thesis can be used as a guide to implementing the first prototypes of the theatre robotics platform. The implementation should answer some of the multidisciplinary needs in the educational field.

## 6.1 Research answers

The research presented in this thesis was conducted to answer the following research questions.

**Q1:** What does the term *theatre robotics* mean?

Theatre robotics is a platform which instantiates *robotic storytelling*. The term *robotic storytelling* was introduced to support the concept development of the theatre robotics environment. Robotic storytelling adds a new layer to digital storytelling by enhancing it with robotics. The theatre robotics platform consists of different modules, which built up a framework which can be used to implement theatre performances with robots. Theatre robotics aims to answer a need of truly multidisciplinary teaching environment in the educational field by merging several school subjects, such as physics, engineering, arts, computer science, literature and history. Different skills and knowledge from these subjects are needed in order to successfully implement a play with the theatre robotics platform.

**Q2:** Which kind of environment the platform should be designed for?

The theatre robotics platform is designed for school classes and especially children's after school clubs, such as SciKids' technology club. Therefore, the design of the platform was highly influenced by the experiences from SciKids' technology clubs, where I have been working as an instructor for several years. The next step with theatre robotics would be implementing this architecture and SciKids' clubs could be used as a primary testing context.

**Q3:** What are the requirements for the platform in sense of hardware and software?

Requirements for the platform were acquired from the analysis of the experiences gathered from SciKids' clubs. The most essential requirements are following: durability and versatility of the toolkit, the accessibility of the programming environment, price-tier of the platform, and capability to position the robots in adequate accuracy. These requirements were answered by choosing affordable open source components, designing new suitable approaches for programming, and sketching an indoor positioning system.

**Q4:** Does the complete design meet the requirements?

The evaluation with criteria used in similar projects of a similar kind advocates the created theatre robotics platform design. Grounding the design on the experiences from SciKids' technology clubs gives it accountability, because SciKids' clubs are also target environments of the theatre robotics platform. The created architecture provides enough extendibility and flexibility with high cost-efficiency, so the design seems to be suitable for the cause. The most critical part found out during the development and evaluation was the durability of the used components and it should be considered more in the future. Implementation of the design will answer this question in more detail in future research.

## 6.2 Further questions

After implementing and critically researching optional approaches to durability issues, the future work would be in testing the platform. Testing the platform in real environment will surely rise issues and questions which need more development. For example, is the scripting environment intuitive and should its abstraction level be made higher or lower? Does indoor positioning work on suitable accuracy? How could the durability of the robot building toolkit improved? What are implications of this for K-12 technology education?

# References

[1] Andruseac, G.G., and Iacob, R. Exploring the Potential of Using Educational Robotics as an Effective Tool to Support Collaborative Learning. In *The 4th IEEE International Conference on E-Health and Bioengineering*, 2013.

[2] Arduino SA. Arduino - Home. `http://www.arduino.cc`, 2015. Accessed: 2015-02-03.

[3] Brown, N.C.C., Sentance, S., Crick, T., and Humphreys, S. Restart: The Resurgence of Computer Science in UK Schools. *ACM Transactions on Computing Education*, 2014.

[4] Brunvand, E., and Stout, P. Kinetic art and embedded systems: a natural collaboration. In *Proceedings of the 42nd ACM technical syumposium on Computer science education*, pages 323–328, 2011.

[5] Caldwell, E.R., and Jones, E.J. Beyond wrestling: using sumobots to engage students in the computer science classroom. *Journal of Computing Sciences in Colleges*, 29(2):132–139, 2013.

[6] Duveskog, M. *Digital Storytelling for HIV and AIDS Education in Africa*. PhD thesis, University of Eastern Finland, 2015.

[7] Ecma International. ECMA-404 - The JSON Data Interchange Format. Technical report, Ecma International, 2013. `http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf`.

[8] Eronen, P.J., Jormanainen, I., Sutinen, E., and Virnes, M. Kids' Club Reborn: Evolution of Activities. In *The 5th IEEE International Conference on Advanced Learning Technology*, volume ICALT 2005, pages 545–547, 2005.

[9] Eronen, P.J., Sutinen, E., Vesisenaho, M., and Virnes, M. Kids' Club as an ICT-based Learning Laboratory. In *Informatics in Education 1*, pages 61–72, 2002.

[10] García Sierra, J.F., Rodríguez Lera, F.J., Fernández Llamas, C., and Matellán Olivera, V. Inside the maze: who would find the cheese first, a robot or a mouse?: teaching IT using robots. In *Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality*, pages 297–302, 2013.

[11] Hu, M., Winikoff, M., and Cranefield, S. Teaching Novice Programming Usign Goals and Plans in a Visual Notation. In *Proceedings of the Fourteenth Australasian Computing Education Conference*, pages 43–52, 2012.

[12] Jormanainen, I. *Supporting Teachers in Unpredictable Robotics Learning Environments*. PhD thesis, University of Eastern Finland, 2013.

[13] Koulouri, T., Lauria, S., and Macredie, R.D. Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches. *ACM Transactions on Computing Education*, 14(4), December 2014.

[14] Laurel, B. *Computers as Theatre*. Addison-Wesley Professional, 2014.

[15] Lönnrot, E. *Kalevala*. J. C. Frenckellin ja Poika, 1849.

[16] McKenney, S., and Reeves, T.C. Educational Design Research. In *Handbook of Research on Educational Communications and Technology, Spector, J.M., Merrill, M.D., Elen, J., and Bishop, M.J.*, pages 131–140. Springer New York, 2014.

[17] MECCANO. Official website Meccano®. `http://www.meccano.com`, 2015. Accessed: 2015-01-15.

[18] Nocs, L. *The Robot: The Life Story of a Technology*. The Johns Hopkins University Press, 2008.

[19] Oksar, I. A Bluetooth signal strength based indoor localization method. In *International Conference on Systems, Signals and Image Processing (IWSSIP)*.

[20] RepRap. G-code - RepRapWiki. `http://reprap.org/wiki/G-code`, 2015. Accessed: 2015-02-10.

[21] Rick, D., Morisse, M., and Schirmer, I. Bringing contexts into the classroom: a design-based approach. In *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, volume ACM, pages 105–115, 2012.

[22] Sadik, A. Digital storytelling: a meaningful technology-integrated approach for engaged student learning. *Educational Technology Research and Development*, 56(4):487–506, 2008.

[23] Salgian, A., Nakra, T.M., Ault, C., and Wang, Y. Conducting robots- Bringing the gap between science, technology and the arts in the undergraduate curriculum. In *Integrated STEM Education Conference (ISEC)*, pages 1–4, 2013.

[24] Stathakis, A., and Petriu, E.M. Robust Pseudo-Random Fiducial Marker for Indoor Localization. In *IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2011.

[25] The BeagleBoard.org Foundation. BeagleBoard.org - community supported open hardware computers for making. `http://beagleboard.org`, 2015. Accessed: 2015-01-05.

[26] The LEGO Group. Home - Mindstorms LEGO.com. `http://www.lego.com/en-us/mindstorms`, 2015. Accessed: 2015-02-11.

[27] The Raspberry Pi Foundation. Raspberry Pi. `http://www.raspberrypi.org`, 2015. Accessed: 2015-01-05.

[28] Theater robotics for human technology education: Implementation issues. Jormanainen, I., Laamanen, M., and Sutinen, E. unpublished, 2015.

[29] Tsang, E., Gavan, C., and Anderson, M. The Practical Application of LEGO®MINDSTORMS®Robotics Kit: Does it Enhance Undergraduate Computing Students' Engagement in Learning the Java Programming Language? In *Proceedings of the 15th Annual Conference on Information technology education*, 2014.

[30] Weiss, R., and Overcast, I. Finding your bot-mate: criteria for evaluating robot kits for use in undergraduate computer science education. *Journal of Computing Sciences in Colleges*, 24(2):43–49, December 2008.

[31] Williams, A. The qualitative impact of using LEGO MIDSTORMS robots to teach computer engineering. *IEEE Transactions on Education*, 46(206), 2003.

[32] Zhang, Y., Kinshuk, Jormannainen, I., and Sutinen, E. An Implementation of the Agency Architecture in Educational Robotics. In *Eighth IEEE International Conference on Advanced Learning Technologies*, pages 194–198, 2008.